

Verwendung von Cut

Betrachte folgendes Programm zum Sortieren einer Liste:

```
% msort(Xs, Ys) Ys ist die Liste Xs sortiert
msort(Xs, Ys) :-
    append(As, [X, Y | Bs], Xs),
    X > Y,
    append(As, [Y, X | Bs], Xs1),
    msort(Xs1, Ys).
msort(Xs, Xs) :- ordered(Xs).

% ordered(Xs) Xs ist eine sortierte Liste
ordered([_]).
ordered([X, Y | Ys]) :- X =< Y, ordered([Y | Ys]).

% append gem. Beilagen (Exhaustive Searching p. 176)
```

Feststellung:

- Es gibt mehrere Arten eine Liste zu sortieren. Dies hängt davon ab, welche adjazenten Elemente X und Y das erste `append` findet.
- Eine sortierte Liste ist immer eindeutig bestimmt, unabhängig vom Rechnungsweg.

Dasselbe Programm mit `cut`, zum verhindern von Backtrack.

```
msort(Xs, Ys) :-
    append(As, [X, Y | Bs], Xs),
    X > Y,
    !,
    append(As, [Y, X | Bs], Xs1),
    msort(Xs1, Ys).
msort(Xs, Xs) :- ordered(Xs), !.
```

Feststellung:

- Jetzt wird nur eine Lösung gefunden, der Rechnungsweg ist durch `cut` fixiert.
- Das deklarative Verhalten ändert sich nicht.
- Die Rekursionsverankerung wird nur erreicht, wenn keine adjazenten Elemente X und Y gefunden werden, die nicht sortiert sind. Die Abfrage `ordered` kann also weggelassen werden. Jetzt ist aber der `cut` in `msort` wichtig: ohne `cut` und `ordered` würde Prolog so antworten:

```

msort(Xs, Ys) :-
    append(As, [X, Y | Bs], Xs),
    X > Y,
    append(As, [Y, X | Bs], Xs1),
    msort(Xs1, Ys).
msort(Xs, Xs).

```

```

| ?- msort([3,1,2], [3,1,2]).
    yes
| ?-

```

In obigem, falschem Programm sollte uns die Zeile `msort(Xs, Xs)` stutzig machen. Sie bedeutet ja, dass **jede** Liste `Xs` sortiert ist.

Die korrekte, effiziente Sortierart muss also lauten:

```

msort(Xs, Ys) :-
    append(As, [X, Y | Bs], Xs),
    X > Y,
    !,                                     % cut aendert dekl. Bedeutung
    append(As, [Y, X | Bs], Xs1),
    msort(Xs1, Ys).
msort(Xs, Xs) :- !.

```

Ein solcher cut, ändert die deklarative Bedeutung des Programmes. Man nennt solche cuts auch *red cuts*, im Gegensatz zu *green cuts*. Letztere ändern die deklarative Bedeutung eines Programmes nicht. Durch Wegnehmen von `ordered` wurde aus dem green cut der zweiten Version ein Roter.